Question 1. (4-4-4-2) Using the following definition of the derivative and the definition of the Heaviside step function :

$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} \qquad H(x) = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

1. Show that the derivative of the rectified linear unit $g(x) = \max\{0, x\}$, **wherever it exists**, is equal to the Heaviside step function.
2. Give two alternative definitions of $g(x)$ using $H(x)$.
3. Show that $H(x)$ can be well approximated by the sigmoid function $\sigma(x) = \frac{1}{1+e^{-kx}}$ asymptotically (i.e for large k), where k is a parameter.

*4. Although the Heaviside step function is not differentiable, we can define its **distributional derivative**. For a function F, consider the functional $F[\phi] = \int_{\mathbb{R}} F(x)\phi(x)dx$, where ϕ is a smooth function (infinitely differentiable) with compact support ($\phi(x) = 0$ whenever $|x| \geq A$, for some $A > 0$).

Show that whenever F is differentiable, $F'[\phi] = -\int_{\mathbb{R}} F(x)\phi'(x)dx$. Using this formula as a definition in the case of non-differentiable functions, show that $H'[\phi] = \phi(0)$. ($\delta[\phi] \doteq \phi(0)$ is known as the Dirac delta function.)

Answer 1. 1. For $x > 0$:

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon} = \lim_{\epsilon \to 0} \frac{x+\epsilon - x}{\epsilon} = 1$$

For $x < 0$ (such that $x + \epsilon < 0$ as $\epsilon \to 0$) :

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon} = \lim_{\epsilon \to 0} \frac{0 - 0}{\epsilon} = 0$$

For $x = 0$, approaching from the left gives 0, while approaching from the right gives 1, hence the derivative of $g(x)$ is not defined at $x = 0$.

$\therefore$ The derivative of $g(x)$ for $x \neq 0$ is equal to $H(x)$.

2. (a) $g(x) = x * H(x)$
   (b) $g(x) = x * (1 - H(-x))$

3. For large k, for x > 0 :

$$\lim_{k \to +\infty} \sigma(x) = \lim_{k \to +\infty} \frac{1}{1+e^{-kx}} = \lim_{k \to +\infty} \frac{1}{1+e^{<large\ negative\ number>}} = 1$$

For large k, for x < 0 :

$$\lim_{k \to +\infty} \sigma(x) = \lim_{k \to +\infty} \frac{1}{1+e^{-kx}} = \lim_{k \to +\infty} \frac{1}{1+e^{<large\ positive\ number>}} = 0$$

$\therefore H(x)$ can be well approximated by the sigmoid function $\sigma(x)$ asymptotically.

4. $F'[\phi] = \int_{\mathbb{R}} F'(x)\phi(x)dx$
   Performing this integration by parts :
   $F'[\phi] = \int_{\mathbb{R}} F'(x)\phi(x)dx = F(x)\phi(x)|_{-\infty}^{+\infty} \ - \ \int_{\mathbb{R}} F(x)\phi'(x)dx$
   Since $\phi(x)$ has a compact support, $F(x)\phi(x)|_{-\infty}^{+\infty} = 0 - 0 = 0$.
   $\therefore F'[\phi] = -\int_{\mathbb{R}} F(x)\phi'(x)dx$ (whenever F is differentiable).

   Thus, $H'(\phi) = -\int_{\mathbb{R}} H(x)\phi'(x)dx = -\int_0^{+\infty} \phi'(x)dx = -\phi(x)|_0^{+\infty} = \phi(0)$ (again, since $\phi(x)$ has a compact support).
   $\therefore H'(\phi) = \phi(0)$

Question 2. (5-8-5-5) Let $\boldsymbol{x}$ be an n-dimensional vector. Recall the softmax function : $S : \boldsymbol{x} \in \mathbb{R}^n \mapsto S(\boldsymbol{x}) \in \mathbb{R}^n$ such that $S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$; the diagonal function : $\text{diag}(\boldsymbol{x})_{ij} = \boldsymbol{x}_i$ if $i = j$ and $\text{diag}(\boldsymbol{x})_{ij} = 0$ if $i \neq j$; and the Kronecker delta function : $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

1. Show that the derivative of the softmax function is $\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} = S(\boldsymbol{x})_i\,(\delta_{ij} - S(\boldsymbol{x})_j)$.
2. Express the Jacobian matrix $\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}$ using matrix-vector notation. Use $\text{diag}(\cdot)$.
3. Compute the Jacobian of the sigmoid function $\sigma(\boldsymbol{x}) = 1/(1 + e^{-\boldsymbol{x}})$.
4. Let $\boldsymbol{y}$ and $\boldsymbol{x}$ be $n-$dimensional vectors related by $\boldsymbol{y} = f(\boldsymbol{x})$, L be an unspecified differentiable loss function. According to the chain rule of calculus, $\nabla_{\boldsymbol{x}} L = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L$, which takes up $\mathcal{O}(n^2)$ computational time in general. Show that if $f(\boldsymbol{x}) = \sigma(\boldsymbol{x})$ or $f(\boldsymbol{x}) = S(\boldsymbol{x})$, the above matrix-vector multiplication can be simplified to a $\mathcal{O}(n)$ operation.

Answer 2. 1. To show that : $\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} = S(\boldsymbol{x})_i\,(\delta_{ij} - S(\boldsymbol{x})_j)$
   Using quotient rule :

$$\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} = \frac{d}{d\boldsymbol{x}_j}\left(\frac{e^{\boldsymbol{x}_i}}{\sum_k e^{\boldsymbol{x}_k}}\right) = \frac{\sum_k e^{\boldsymbol{x}_k} . \frac{d}{d\boldsymbol{x}_j}(e^{\boldsymbol{x}_i}) - e^{\boldsymbol{x}_i} . \frac{d}{d\boldsymbol{x}_j}(\sum_k e^{\boldsymbol{x}_k})}{(\sum_k e^{\boldsymbol{x}_k})^2}$$

   Here, $\frac{d}{d\boldsymbol{x}_j}(e^{\boldsymbol{x}_i}) = \begin{cases} e^{\boldsymbol{x}_i} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \implies \frac{d}{d\boldsymbol{x}_j}(e^{\boldsymbol{x}_i}) = \delta_{ij}.e^{\boldsymbol{x}_i}$
   Also, $\frac{d}{d\boldsymbol{x}_j}(\sum_k e^{\boldsymbol{x}_k}) = e^{\boldsymbol{x}_j}$
   Thus,

$$\begin{aligned}
\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} &= \frac{\sum_k e^{\boldsymbol{x}_k}.\delta_{ij}.e^{\boldsymbol{x}_i} - e^{\boldsymbol{x}_i}.e^{\boldsymbol{x}_j}}{(\sum_k e^{\boldsymbol{x}_k})^2} \\
&= \frac{e^{\boldsymbol{x}_i}(\sum_k e^{\boldsymbol{x}_k}.\delta_{ij} - e^{\boldsymbol{x}_j})}{(\sum_k e^{\boldsymbol{x}_k})^2} \\
&= \frac{e^{\boldsymbol{x}_i}}{\sum_k e^{\boldsymbol{x}_k}}.\left(\frac{\sum_k e^{\boldsymbol{x}_k}.\delta_{ij}}{\sum_k e^{\boldsymbol{x}_k}} - \frac{e^{\boldsymbol{x}_j}}{\sum_k e^{\boldsymbol{x}_k}}\right) \\
&= S(\boldsymbol{x})_i\,(\delta_{ij} - S(\boldsymbol{x})_j)
\end{aligned}$$

   $\therefore \frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} = S(\boldsymbol{x})_i\,(\delta_{ij} - S(\boldsymbol{x})_j)$

2. Jacobian of softmax :

$$\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial S(\boldsymbol{x})_1}{\partial \boldsymbol{x}_1} & \frac{\partial S(\boldsymbol{x})_1}{\partial \boldsymbol{x}_2} & \dots & \frac{\partial S(\boldsymbol{x})_1}{\partial \boldsymbol{x}_n} \\ \frac{\partial S(\boldsymbol{x})_2}{\partial \boldsymbol{x}_1} & \frac{\partial S(\boldsymbol{x})_2}{\partial \boldsymbol{x}_2} & \dots & \frac{\partial S(\boldsymbol{x})_2}{\partial \boldsymbol{x}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S(\boldsymbol{x})_n}{\partial \boldsymbol{x}_1} & \frac{\partial S(\boldsymbol{x})_n}{\partial \boldsymbol{x}_2} & \dots & \frac{\partial S(\boldsymbol{x})_n}{\partial \boldsymbol{x}_n} \end{bmatrix}$$

$$= \begin{bmatrix} S(\boldsymbol{x})_1.(1 - S(\boldsymbol{x})_1) & -S(\boldsymbol{x})_1.S(\boldsymbol{x})_2 & \dots & -S(\boldsymbol{x})_1.S(\boldsymbol{x})_n \\ -S(\boldsymbol{x})_2.S(\boldsymbol{x})_1 & S(\boldsymbol{x})_2.(1 - S(\boldsymbol{x})_2) & \dots & -S(\boldsymbol{x})_2.S(\boldsymbol{x})_n \\ \vdots & \vdots & \ddots & \vdots \\ -S(\boldsymbol{x})_n.S(\boldsymbol{x})_1 & -S(\boldsymbol{x})_n.S(\boldsymbol{x})_2 & \dots & S(\boldsymbol{x})_n.(1 - S(\boldsymbol{x})_n) \end{bmatrix}$$

$$= \begin{bmatrix} S(\boldsymbol{x})_1 - S(\boldsymbol{x})_1.S(\boldsymbol{x})_1 & -S(\boldsymbol{x})_1.S(\boldsymbol{x})_2 & \dots & -S(\boldsymbol{x})_1.S(\boldsymbol{x})_n \\ -S(\boldsymbol{x})_2.S(\boldsymbol{x})_1 & S(\boldsymbol{x})_2 - S(\boldsymbol{x})_2.S(\boldsymbol{x})_2 & \dots & -S(\boldsymbol{x})_2.S(\boldsymbol{x})_n \\ \vdots & \vdots & \ddots & \vdots \\ -S(\boldsymbol{x})_n.S(\boldsymbol{x})_1 & -S(\boldsymbol{x})_n.S(\boldsymbol{x})_2 & \dots & S(\boldsymbol{x})_n - S(\boldsymbol{x})_n.S(\boldsymbol{x})_n \end{bmatrix}$$

$$= \begin{bmatrix} S(\boldsymbol{x})_1 & 0 & \dots & 0 \\ 0 & S(\boldsymbol{x})_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S(\boldsymbol{x})_n \end{bmatrix} - \begin{bmatrix} S(\boldsymbol{x})_1.S(\boldsymbol{x})_1 & S(\boldsymbol{x})_1.S(\boldsymbol{x})_2 & \dots & S(\boldsymbol{x})_1.S(\boldsymbol{x})_n \\ S(\boldsymbol{x})_2.S(\boldsymbol{x})_1 & S(\boldsymbol{x})_2.S(\boldsymbol{x})_2 & \dots & S(\boldsymbol{x})_2.S(\boldsymbol{x})_n \\ \vdots & \vdots & \ddots & \vdots \\ S(\boldsymbol{x})_n.S(\boldsymbol{x})_1 & S(\boldsymbol{x})_n.S(\boldsymbol{x})_2 & \dots & S(\boldsymbol{x})_n.S(\boldsymbol{x})_n \end{bmatrix}$$

$$= \text{diag}(S(\boldsymbol{x})) - \begin{bmatrix} S(\boldsymbol{x})_1 \\ S(\boldsymbol{x})_2 \\ \vdots \\ S(\boldsymbol{x})_n \end{bmatrix} . \begin{bmatrix} S(\boldsymbol{x})_1 & S(\boldsymbol{x})_2 & \dots & S(\boldsymbol{x})_n \end{bmatrix}$$

$$= \text{diag}(S(\boldsymbol{x})) - S(\boldsymbol{x}).S(\boldsymbol{x})^\top$$

3. $\frac{d}{d\boldsymbol{x}_i}\sigma(\boldsymbol{x})_i = \frac{d}{d\boldsymbol{x}_i}\left(\frac{1}{1+e^{-\boldsymbol{x}_i}}\right) = -\frac{-e^{-\boldsymbol{x}_i}}{(1+e^{-\boldsymbol{x}_i})^2} = \frac{1}{1+e^{-\boldsymbol{x}_i}}.\frac{e^{-\boldsymbol{x}_i}+1-1}{1+e^{-\boldsymbol{x}_i}} = \frac{1}{1+e^{-\boldsymbol{x}_i}}.\left(\frac{1+e^{-\boldsymbol{x}_i}}{1+e^{-\boldsymbol{x}_i}} - \frac{1}{1+e^{-\boldsymbol{x}_i}}\right)$

$= \sigma(\boldsymbol{x})_i * (1 - \sigma(\boldsymbol{x})_i)$

$\frac{d}{d\boldsymbol{x}_j}\sigma(\boldsymbol{x})_i = 0$

$\therefore$ Jacobian of sigmoid :

$$\frac{\partial \sigma(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \sigma(\boldsymbol{x})_1}{\partial \boldsymbol{x}_1} & \frac{\partial \sigma(\boldsymbol{x})_1}{\partial \boldsymbol{x}_2} & \dots & \frac{\partial \sigma(\boldsymbol{x})_1}{\partial \boldsymbol{x}_n} \\ \frac{\partial \sigma(\boldsymbol{x})_2}{\partial \boldsymbol{x}_1} & \frac{\partial \sigma(\boldsymbol{x})_2}{\partial \boldsymbol{x}_2} & \dots & \frac{\partial \sigma(\boldsymbol{x})_2}{\partial \boldsymbol{x}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \sigma(\boldsymbol{x})_n}{\partial \boldsymbol{x}_1} & \frac{\partial \sigma(\boldsymbol{x})_n}{\partial \boldsymbol{x}_2} & \dots & \frac{\partial \sigma(\boldsymbol{x})_n}{\partial \boldsymbol{x}_n} \end{bmatrix}$$

$$= \begin{bmatrix} \sigma(\boldsymbol{x})_1 * (1 - \sigma(\boldsymbol{x})_1) & 0 & \dots & 0 \\ 0 & \sigma(\boldsymbol{x})_2 * (1 - \sigma(\boldsymbol{x})_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma(\boldsymbol{x})_n * (1 - \sigma(\boldsymbol{x})_n) \end{bmatrix}$$

$$= \text{diag}(\sigma(\boldsymbol{x}) * (1 - \sigma(\boldsymbol{x})))$$

4. $\nabla_{\boldsymbol{x}} L = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L$
   If $\boldsymbol{y} = \sigma(\boldsymbol{x})$, $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ is the Jacobian of sigmoid $= \text{diag}(\sigma(\boldsymbol{x}) * (1 - \sigma(\boldsymbol{x})))$. Since this is a diagonal matrix, it is sufficient to compute only the n diagonal elements, and multiply each of them with the corresponding element in $\nabla_{\boldsymbol{y}} L$. Hence, this can be done in $\mathcal{O}(n)$ time.

   If $\boldsymbol{y} = S(\boldsymbol{x})$, $(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{x}} L = \left(\text{diag}(S(\boldsymbol{x})) - S(\boldsymbol{x}).S(\boldsymbol{x})^\top\right) \nabla_{\boldsymbol{x}} L = \text{diag}(S(\boldsymbol{x})) \nabla_{\boldsymbol{x}} L - S(\boldsymbol{x}).S(\boldsymbol{x})^\top . \nabla_{\boldsymbol{x}} L$
   We just saw that $\text{diag}(S(\boldsymbol{x})) \nabla_{\boldsymbol{x}} L$ can be computed in $\mathcal{O}(n)$ time. $S(\boldsymbol{x})^\top . \nabla_{\boldsymbol{x}} L$ is an inner product of two n-dimensional vectors, and so takes $\mathcal{O}(n)$ time to produce a scalar. $S(\boldsymbol{x}). <$ scalar $>$ also takes $\mathcal{O}(n)$ time. Hence, $\nabla_{\boldsymbol{x}} L$ takes $\mathcal{O}(n)$ time even in case of $\boldsymbol{y} = S(\boldsymbol{x})$.

Question 3. (3-3-3-3) Recall the definition of the softmax function : $S(\boldsymbol{x})_i = e^{\boldsymbol{x}_i} / \sum_j e^{\boldsymbol{x}_j}$.

1. Show that softmax is translation-invariant, that is : $S(\boldsymbol{x} + c) = S(\boldsymbol{x})$, where c is a scalar constant.
2. Show that softmax is not invariant under scalar multiplication. Let $S_c(\boldsymbol{x}) = S(c\boldsymbol{x})$ where $c \geq 0$. What are the effects of taking c to be 0 and arbitrarily large ?
3. Let $\boldsymbol{x}$ be a 2-dimentional vector. One can represent a 2-class categorical probability using softmax $S(\boldsymbol{x})$. Show that $S(\boldsymbol{x})$ can be reparameterized using sigmoid function, i.e. $S(\boldsymbol{x}) = [\sigma(z), 1 - \sigma(z)]^\top$ where z is a scalar function of $\boldsymbol{x}$.
4. Let $\boldsymbol{x}$ be a K-dimentional vector ($K \geq 2$). Show that $S(\boldsymbol{x})$ can be represented using $K - 1$ parameters, i.e. $S(\boldsymbol{x}) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$ where y_i is a scalar function of $\boldsymbol{x}$ for $i \in \{1, ..., K - 1\}$.

Answer 3. 1. $S(\boldsymbol{x} + c)_i = \frac{e^{(\boldsymbol{x}_i + c)}}{\sum_j e^{(\boldsymbol{x}_j + c)}} = \frac{e^c * e^{\boldsymbol{x}_i}}{\sum_j e^c * e^{(\boldsymbol{x}_j)}} = \frac{e^c * e^{\boldsymbol{x}_i}}{e^c * \sum_j e^{(\boldsymbol{x}_j)}} = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{(\boldsymbol{x}_j)}} = S(\boldsymbol{x})_i$
$\implies S(\boldsymbol{x} + c) = S(\boldsymbol{x})$

2. $S_c(\boldsymbol{x})_i = S(c\boldsymbol{x}) = \frac{e^{c\boldsymbol{x}_i}}{\sum_j e^{c\boldsymbol{x}_j}} = \frac{(e^{\boldsymbol{x}_i})^c}{\sum_j (e^{\boldsymbol{x}_j})^c} \neq S(\boldsymbol{x})_i$ unless $c = 1$. Hence, softmax is not invariant under scalar multiplication.

$$c = 0 \implies S_c(\boldsymbol{x})_i = \frac{(e^{\boldsymbol{x}_i})^0}{\sum_j (e^{\boldsymbol{x}_j})^0} = \frac{1}{\sum_j 1} = \frac{1}{n}$$

   Hence, when $c = 0$, the softmax values in every dimension are equal to $1/n$.

$$c \to +\infty \implies S_c(\boldsymbol{x})_i = \lim_{c \to +\infty} \frac{(e^{\boldsymbol{x}_i})^c}{\sum_j (e^{\boldsymbol{x}_j})^c} = \lim_{c \to +\infty} \frac{1}{\sum_j (\frac{e^{\boldsymbol{x}_j}}{e^{\boldsymbol{x}_i}})^c}$$

   If $\boldsymbol{x}_i = \boldsymbol{x}_j$, $\lim_{c \to +\infty} (\frac{e^{\boldsymbol{x}_j}}{e^{\boldsymbol{x}_i}})^c = 1$.
   If $\boldsymbol{x}_i > \boldsymbol{x}_j$, $\lim_{c \to +\infty} (\frac{e^{\boldsymbol{x}_j}}{e^{\boldsymbol{x}_i}})^c = 0$.
   If $\boldsymbol{x}_i < \boldsymbol{x}_j$, $\lim_{c \to +\infty} (\frac{e^{\boldsymbol{x}_j}}{e^{\boldsymbol{x}_i}})^c \to +\infty$.
   So, if $\boldsymbol{x}_i$ is the maximum of all $\boldsymbol{x}_j$s, $\lim_{c \to +\infty} 1 / \sum_j (\frac{e^{\boldsymbol{x}_j}}{e^{\boldsymbol{x}_i}})^c = 1/(0 + 0 + \cdots + 1 + \cdots + 0) = 1$. For any other $\boldsymbol{x}_i$, $\lim_{c \to +\infty} 1 / \sum_j (\frac{e^{\boldsymbol{x}_j}}{e^{\boldsymbol{x}_i}})^c = 1/(\infty + \infty + \cdots + 1 + \cdots + \infty) = 0$.
   Hence, when $c \to +\infty$, the output is 1 at the dimension with the highest value, and 0 in all other dimensions.

3. When $\boldsymbol{x}$ is 2-dimensional, say $[x_1, x_2]^\top$,
   $S(\boldsymbol{x})_1 = S(x_1) = \frac{e^{x1}}{e^{x1} + e^{x2}} = \frac{1}{1 + e^{x2 - x1}} = \frac{1}{1 + e^{-(x1 - x2)}} = \sigma(x1 - x2)$

$S(\boldsymbol{x})_2 = S(x_2) = \frac{e^{x2}}{e^{x1}+e^{x2}} = \frac{1}{e^{x1-x2}+1} = \frac{1+e^{x1-x2}-e^{x1-x2}}{e^{x1-x2}+1} = 1 - \frac{e^{x1-x2}}{e^{x1-x2}+1} = 1 - \frac{1}{1+e^{-(x1-x2)}}$
$= 1 - \sigma(x1 - x2)$
Hence, if $z = x1 - x2$, $S(\boldsymbol{x}) = [\sigma(z), 1 - \sigma(z)]^\top$.

4. Let $\boldsymbol{x} = [x_1, x_2, \cdots, x_K]^\top$.
$S(\boldsymbol{x}) = [S(x_1), S(x_2), \cdots, S(x_K)]^\top$
$S(\boldsymbol{x})_1 = S(x_1) = \frac{e^{x_1}}{e^{x_1}+e^{x_2}+\cdots+e^{x_K}} = \frac{1}{1+e^{(x_2-x_1)}+\cdots+e^{(x_K-x_1)}} = \frac{e^0}{e^0+e^{(x_2-x_1)}+\cdots+e^{(x_K-x_1)}}$
$S(\boldsymbol{x})_j = S(x_j) = \frac{e^{x_j}}{e^{x_1}+\cdots+e^{x_j}+\cdots} = \frac{e^{(x_j-x_1)}}{e^0+\cdots+e^{(x_j-x_1)}+\cdots}$
Thus, we can see that $S(\boldsymbol{x}) = S([x_1, x_2, x_3, \cdots, x_K]^\top) = S([0, x_2 - x_1, x_3 - x_1, \cdots, x_K - x_1])$.
Hence, $S(\boldsymbol{x})$ can be represented using $K - 1$ parameters.

Question 4. (15) Consider a 2-layer neural network $y : \mathbb{R}^D \to \mathbb{R}^K$ of the form :

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)}$$

for $1 \leq k \leq K$, with parameters $\Theta = (\omega^{(1)}, \omega^{(2)})$ and logistic sigmoid activation function σ. Show that there exists an equivalent network of the same form, with parameters $\Theta' = (\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)})$ and tanh activation function, such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ for all $x \in \mathbb{R}^D$, and express Θ' as a function of Θ.

Answer 4. Recall that $\sigma(x) = \frac{1}{1+e^{-x}}$, and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

$\tanh(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}} = \frac{1-e^{-2x}}{1+e^{-2x}} = \frac{1+1-1-e^{-2x}}{1+e^{-2x}} = 2.\frac{1}{1+e^{-2x}} - 1 = 2\sigma(2x) - 1$

$\implies \sigma(x) = \frac{1}{2}\left(\tanh(\frac{x}{2}) + 1\right)$

$$\begin{aligned}
y(x, \Theta, \sigma)_k &= \sum_{j=1}^{M} \omega_{kj}^{(2)} .\ \sigma \left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)} \\
&= \sum_{j=1}^{M} \omega_{kj}^{(2)} .\ \frac{1}{2} \left(\tanh \left(\frac{1}{2} (\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}) \right) + 1 \right) \ + \ \omega_{k0}^{(2)} \\
&= \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} \left(\tanh \left(\sum_{i=1}^{D} \frac{\omega_{ji}^{(1)}}{2} x_i + \frac{\omega_{j0}^{(1)}}{2} \right) + 1 \right) \ + \ \omega_{k0}^{(2)} \\
&= \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} \ \tanh \left(\sum_{i=1}^{D} \frac{\omega_{ji}^{(1)}}{2} x_i + \frac{\omega_{j0}^{(1)}}{2} \right) \ + \ \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} + \omega_{k0}^{(2)} \\
&= \sum_{j=1}^{M} \tilde{\omega}_{kj}^{(2)} \tanh \left(\sum_{i=1}^{D} \tilde{\omega}_{ji}^{(1)} x_i + \tilde{\omega}_{j0}^{(1)} \right) + \tilde{\omega}_{k0}^{(2)} \\
&= y(x, \Theta', \tanh)
\end{aligned}$$

$\therefore$ There exists an equivalent network such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ for all $x \in \mathbb{R}^D$.

Here, $\Theta' = \left(\tilde{\omega}^{(1)}, \left(\tilde{\omega}_{k0}^{(2)}, \tilde{\omega}_{k1}^{(2)}, \tilde{\omega}_{k2}^{(2)}, ..., \tilde{\omega}_{kM}^{(2)} \right) \right) = \left(\frac{\omega^{(1)}}{2}, \left(\sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} + \omega_{k0}^{(2)},\ \frac{1}{2}\omega_{k1},\ \frac{1}{2}\omega_{k2}, ..., \frac{1}{2}\omega_{kM} \right) \right)$.

Question 5. (2-2-2-2) Given $N \in \mathbb{Z}^+$, we want to show that for any $f : \mathbb{R}^n \to \mathbb{R}^m$ and any sample set $\mathcal{S} \subset \mathbb{R}^n$ of size N, there is a set of parameters for a two-layer network such that the output $y(\boldsymbol{x})$ matches $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$. That is, we want to interpolate f with y on any finite set of samples $\mathcal{S}$.

1. Write the generic form of the function $y : \mathbb{R}^n \to \mathbb{R}^m$ defined by a 2-layer network with $N-1$ hidden units, with linear output and activation function ϕ, in terms of its weights and biases $(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)})$ and $(\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)})$.
2. In what follows, we will restrict $\boldsymbol{W}^{(1)}$ to be $\boldsymbol{W}^{(1)} = [\boldsymbol{w}, \cdots, \boldsymbol{w}]^\top$ for some $\boldsymbol{w} \in \mathbb{R}^n$ (so the rows of $\boldsymbol{W}^{(1)}$ are all the same). Show that the interpolation problem on the sample set $\mathcal{S} = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}\} \subset \mathbb{R}^n$ can be reduced to solving a matrix equation : $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$, where $\tilde{\boldsymbol{W}}^{(2)}$ and $\boldsymbol{F}$ are both $N \times m$, given by
$$\tilde{\boldsymbol{W}}^{(2)} = [\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}]^\top \qquad \boldsymbol{F} = [f(\boldsymbol{x}^{(1)}), \cdots, f(\boldsymbol{x}^{(N)})]^\top$$
Express the $N \times N$ matrix $\boldsymbol{M}$ in terms of $\boldsymbol{w}$, $\boldsymbol{b}^{(1)}$, ϕ and $\boldsymbol{x}^{(i)}$.
3. *3. **Proof with Relu activation.** Assume $\boldsymbol{x}^{(i)}$ are all distinct. Choose $\boldsymbol{w}$ such that $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$ are also all distinct (Try to prove the existence of such a $\boldsymbol{w}$, although this is not required for the assignment - See Assignment 0). Set $\boldsymbol{b}_j^{(1)} = -\boldsymbol{w}^\top\boldsymbol{x}^{(j)} + \epsilon$, where $\epsilon > 0$. Find a value of ϵ such that $\boldsymbol{M}$ is triangular with non-zero diagonal elements. Conclude. (Hint : assume an ordering of $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$.)
4. *4. **Proof with sigmoid-like activations**. Assume ϕ is continuous, bounded, $\phi(-\infty) = 0$ and $\phi(0) > 0$. Decompose $\boldsymbol{w}$ as $\boldsymbol{w} = \lambda\boldsymbol{u}$. Set $\boldsymbol{b}_j^{(1)} = -\lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)}$. Fixing $\boldsymbol{u}$, show that $\lim_{\lambda\to+\infty}\boldsymbol{M}$ is triangular with non-zero diagonal elements. Conclude. (Note that doing so preserves the distinctness of $\boldsymbol{w}^\top\boldsymbol{x}^{(i)}$.)

Answer 5. 1.
$$y(\boldsymbol{x}) = \boldsymbol{W}^{(2)}.\phi(\boldsymbol{W}^{(1)}.\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$$

2. For each $\boldsymbol{x}^{(i)}$ in the sample set $\mathcal{S}$,
$$\begin{aligned} y(\boldsymbol{x}^{(i)}) &= \boldsymbol{W}^{(2)}.\phi(\boldsymbol{W}^{(1)}.\boldsymbol{x}^{(i)} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)} = [\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}].[\phi(\boldsymbol{W}^{(1)}.\boldsymbol{x}^{(i)} + \boldsymbol{b}^{(1)}.1), 1]^\top \\ &= [\phi(\boldsymbol{x}^{(i)\top}.\boldsymbol{W}^{(1)\top} + 1.\boldsymbol{b}^{(1)\top}), 1].[\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}]^\top \end{aligned}$$

Since this is an interpolation problem, we would like $y(\boldsymbol{x}^{(i)}) = f(\boldsymbol{x}^{(i)})$. Combining all $\boldsymbol{x}^{(i)}$s :
$$\begin{aligned} \boldsymbol{F} &= [f(\boldsymbol{x}^{(1)}), \cdots, f(\boldsymbol{x}^{(N)})]^\top = [y(\boldsymbol{x}^{(1)}), \cdots, y(\boldsymbol{x}^{(N)})]^\top \\ &= \begin{bmatrix} y(\boldsymbol{x}^{(1)})^\top \\ \vdots \\ y(\boldsymbol{x}^{(N)})^\top \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{W}^{(2)}.\phi(\boldsymbol{W}^{(1)}.\boldsymbol{x}^{(1)} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}\right)^\top \\ \vdots \\ \left(\boldsymbol{W}^{(2)}.\phi(\boldsymbol{W}^{(1)}.\boldsymbol{x}^{(N)} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}\right)^\top \end{bmatrix} \\ &= \begin{bmatrix} \phi(\boldsymbol{x}^{(1)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}).\boldsymbol{W}^{(2)\top} + \boldsymbol{b}^{(2)\top} \\ \vdots \\ \phi(\boldsymbol{x}^{(N)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}).\boldsymbol{W}^{(2)\top} + \boldsymbol{b}^{(2)\top} \end{bmatrix} \\ &= \begin{bmatrix} \phi(\boldsymbol{x}^{(1)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}) \\ \vdots \\ \phi(\boldsymbol{x}^{(N)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}) \end{bmatrix} .\boldsymbol{W}^{(2)\top} + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} .\boldsymbol{b}^{(2)\top} \end{aligned}$$

$$= \begin{bmatrix} \phi(\boldsymbol{x}^{(1)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}) & 1 \\ \vdots & \vdots \\ \phi(\boldsymbol{x}^{(N)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}) & 1 \end{bmatrix} . \begin{bmatrix} \boldsymbol{W}^{(2)\top} \\ \boldsymbol{b}^{(2)\top} \end{bmatrix} = \boldsymbol{M}.[\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}]^\top = \boldsymbol{M}.\tilde{\boldsymbol{W}}^{(2)}$$

Thus, the interpolation problem can be reduced to solving a matrix equation : $\boldsymbol{M}.\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$. Here, $\boldsymbol{M}$ is :

$$\begin{aligned}
\boldsymbol{M} &= \begin{bmatrix} \phi(\boldsymbol{x}^{(1)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}) & 1 \\ \vdots & \vdots \\ \phi(\boldsymbol{x}^{(N)\top}.\boldsymbol{W}^{(1)\top} + \boldsymbol{b}^{(1)\top}) & 1 \end{bmatrix} \\
&= \begin{bmatrix} \phi(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(1)} + \boldsymbol{b}^{(1)})^\top & 1 \\ \vdots & \vdots \\ \phi(\boldsymbol{W}^{(1)}\boldsymbol{x}^{(N)} + \boldsymbol{b}^{(1)})^\top & 1 \end{bmatrix} \\
&= \begin{bmatrix} \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(1)} + \boldsymbol{b}_1^{(1)}) & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(1)} + \boldsymbol{b}_2^{(1)}) & \cdots & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(1)} + \boldsymbol{b}_{N-1}^{(1)}) & 1 \\ \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(2)} + \boldsymbol{b}_1^{(1)}) & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(2)} + \boldsymbol{b}_2^{(1)}) & \cdots & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(2)} + \boldsymbol{b}_{N-1}^{(1)}) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N-1)} + \boldsymbol{b}_1^{(1)}) & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N-1)} + \boldsymbol{b}_2^{(1)}) & \cdots & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N-1)} + \boldsymbol{b}_{N-1}^{(1)}) & 1 \\ \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N)} + \boldsymbol{b}_1^{(1)}) & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N)} + \boldsymbol{b}_2^{(1)}) & \cdots & \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N)} + \boldsymbol{b}_{N-1}^{(1)}) & 1 \end{bmatrix}
\end{aligned}$$

3. When ϕ is ReLU, and $\boldsymbol{b}_j^{(1)} = -\boldsymbol{w}^\top\boldsymbol{x}^{(j)} + \epsilon$ ($\epsilon > 0$), the diagonal elements of $\boldsymbol{M}$ are :
   $\phi(\boldsymbol{w}^\top\boldsymbol{x}^{(j)} + \boldsymbol{b}_j^{(1)}) = \phi(\epsilon) > 0$ ($\because \epsilon > 0$)
   For $\boldsymbol{M}$ to be triangular, all elements below the diagonal must be negative, so then ReLU non-linearity will make them 0.
   $\implies \boldsymbol{w}^\top\boldsymbol{x}^{(i)} + \boldsymbol{b}_j^{(1)} < 0$ for $i > j \implies \boldsymbol{w}^\top\boldsymbol{x}^{(i)} - \boldsymbol{w}^\top\boldsymbol{x}^{(j)} + \epsilon < 0 \implies \epsilon < \boldsymbol{w}^\top(\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(i)})$
   $\because \epsilon > 0$, we assume that $x^{(i)}$ are ordered such that $\boldsymbol{w}^\top(\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(i)}) > 0 \implies \boldsymbol{w}^\top\boldsymbol{x}^{(j)} > \boldsymbol{w}^\top\boldsymbol{x}^{(i)}$ for $i > j$.
   $\therefore \epsilon$ can be any real number such that :
   $$0 < \epsilon < \min_{i>j} \left(\boldsymbol{w}^\top(\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(i)})\right)$$
   Conclusion : Since M is upper triangular, the linear system $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$ can be solved easily for $\tilde{\boldsymbol{W}}^{(2)}$. The last row of $\boldsymbol{M}$ is simply $[0, \cdots, 0, 1] \implies \text{LastRow}(\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)}) = \boldsymbol{b}^{(2)\top} = \text{LastRow}(\boldsymbol{F}) = f(\boldsymbol{x}^{(N)}\top) \implies \boldsymbol{b}^{(2)} = f(\boldsymbol{x}^{(N)})$.
   Considering second last row : $[0, \cdots, \phi(\boldsymbol{w}^\top\boldsymbol{x}^{(N-1)} + \boldsymbol{b}_{N-1}^{(1)}), 1] = [0, \cdots, \phi(\epsilon), 1] \implies \phi(\epsilon).\boldsymbol{W}_{(N-1)}^{(2)} + \boldsymbol{b}^{(2)} = f(\boldsymbol{x}^{(N-1)}) \implies \boldsymbol{W}_{(N-1)}^{(2)} = (f(\boldsymbol{x}^{(N-1)}) - f(\boldsymbol{x}^{(N)}))/\phi(\epsilon)$. And so on.. Hence, $\tilde{\boldsymbol{W}}^{(2)}$ can be found easily when $\boldsymbol{M}$ is triangular.
4. Decomposing $\boldsymbol{w} = \lambda\boldsymbol{u}$, and setting $\boldsymbol{b}_j^{(1)} = -\lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)}$, $\boldsymbol{M}$ can be formulated as :
   The diagonal elements of $\boldsymbol{M}$ are : $\phi(\boldsymbol{w}^\top\boldsymbol{x}^{(j)} + \boldsymbol{b}_j^{(1)}) = \phi(\lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)} - \lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)}) = \phi(0) > 0$
   The non-diagonal elements are $\phi(\boldsymbol{w}^\top\boldsymbol{x}^{(i)} + \boldsymbol{b}_j^{(1)}) = \phi(\lambda\boldsymbol{u}^\top\boldsymbol{x}^{(i)} - \lambda\boldsymbol{u}^\top\boldsymbol{x}^{(j)}) = \phi\big(\lambda\boldsymbol{u}^\top(x^{(i)} - x^{(j)})\big)$
   $\therefore$ For elements lower than the diagonal, $i > j$. Assuming the same ordering as in 3.,
   $\boldsymbol{w}^\top(\boldsymbol{x}^{(j)} - \boldsymbol{x}^{(i)}) > 0 \implies \lambda\boldsymbol{u}^\top(x^{(i)} - x^{(j)}) < 0$
   $\therefore \lambda \to +\infty \implies$ lower-diagonal elements $\to 0$.
   $\therefore \lim_{\lambda\to+\infty} \boldsymbol{M}$ is triangular with non-zero diagonal elements.
   Conclusion : Same as above, as $\lambda \to +\infty$.

Question 6. (6) Compute the *full*, *valid*, and *same* convolution (with kernel flipping) for the following 1D matrices : $[1, 2, 3, 4] * [1, 0, 2]$

Answer 6. *Full* convolution :

$$[1, 2, 3, 4] * [1, 0, 2] = \begin{bmatrix} [2, 0, 1, 0, 0, 0].[0, 0, 1, 2, 3, 4], \\ [2, 0, 1, 0, 0].[0, 1, 2, 3, 4], \\ [2, 0, 1, 0].[1, 2, 3, 4], \\ [0, 2, 0, 1].[1, 2, 3, 4], \\ [0, 0, 2, 0, 1].[1, 2, 3, 4, 0], \\ [0, 0, 0, 2, 0, 1].[1, 2, 3, 4, 0, 0] \end{bmatrix} = [1, 2, 5, 8, 6, 8]$$

Valid convolution : (don't pad the input)

$$[1, 2, 3, 4] * [1, 0, 2] = \begin{bmatrix} [2, 0, 1, 0].[1, 2, 3, 4], \\ [0, 2, 0, 1].[1, 2, 3, 4] \end{bmatrix} = [5, 8]$$

Same convolution : (pad input only enough to make output the same size)

$$[1, 2, 3, 4] * [1, 0, 2] = \begin{bmatrix} [2, 0, 1, 0, 0].[0, 1, 2, 3, 4], \\ [2, 0, 1, 0].[1, 2, 3, 4], \\ [0, 2, 0, 1].[1, 2, 3, 4], \\ [0, 0, 2, 0, 1].[1, 2, 3, 4, 0] \end{bmatrix} = [2, 5, 8, 6]$$

Question 7. (5-5) Consider a convolutional neural network. Assume the input is a colorful image of size 256×256 in the RGB representation. The first layer convolves 64 8×8 kernels with the input, using a stride of 2 and no padding. The second layer downsamples the output of the first layer with a 5×5 non-overlapping max pooling. The third layer convolves 128 4×4 kernels with a stride of 1 and a zero-padding of size 1 on each border.

1. What is the dimensionality (scalar) of the output of the last layer ?
2. Not including the biases, how many parameters are needed for the last layer ?

Answer 7. 1. Using the formula : $o = \lfloor \frac{i+2p-k}{s} \rfloor + 1$

$$256 \times 256 \times 3 \xrightarrow[64,\ 8\times8,\ s=2,\ p=0]{1} \left(\lfloor \tfrac{256-8}{2} \rfloor + 1\right) \times \left(\lfloor \tfrac{256-8}{2} \rfloor + 1\right) \times 64 = 125 \times 125 \times 64$$

$$125 \times 125 \times 64 \xrightarrow[5\times5 \text{ max pool}]{2} \left(\lfloor \tfrac{125-5}{5} \rfloor + 1\right) \times \left(\lfloor \tfrac{125-5}{5} \rfloor + 1\right) \times 64 = 25 \times 25 \times 64$$

$$25 \times 25 \times 64 \xrightarrow[128,\ 4\times4,\ s=1,\ p=1]{3} \left(\lfloor \tfrac{25+2-4}{1} \rfloor + 1\right) \times \left(\lfloor \tfrac{25+2-4}{1} \rfloor + 1\right) \times 128 = 24 \times 24 \times 128$$

2. Last layer convolves 128 4×4 kernels on a 64-channel input $\implies 128 * (4 * 4 * 64) = 131072$ parameters.

Question 8. (4-4-4) Assume we are given data of size $3 \times 64 \times 64$. In what follows, provide the correct configuration of a convolutional neural network layer that satisfies the specified assumption. Answer with the window size of kernel (k), stride (s), padding (p), and dilation (d, with convention $d = 1$ for no dilation). Use square windows only (i.e. same k for both width and height).

1. The output shape of the first layer is $(64, 32, 32)$.
   (a) Assume $k = 8$ without dilation.
   (b) Assume $d = 7$, and $s = 2$.
2. The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$.
   (a) Specify k and s for pooling with non-overlapping window.
   (b) What is output shape if $k = 8$ and $s = 4$ instead ?
3. The output shape of the last layer is $(128, 4, 4)$.
   (a) Assume we are not using padding or dilation.
   (b) Assume $d = 2$, $p = 2$.
   (c) Assume $p = 1$, $d = 1$.

Answer 8. Using the formulae : $o = \lfloor \frac{i+2p-k'}{s} \rfloor + 1 \,;\, k' = k + (d-1) * (k-1)$

1. (a) $k = 8,\ s = 2,\ p = 3,\ d = 1$
   (b) $k = 1,\ s = 2,\ p = 0,\ d = 7$
2. (a) $k = 4,\ s = 4$
   (b) $64 \times 7 \times 7$
3. (a) $k = 2,\ s = 2,\ p = 0,\ d = 1$
   (b) $k = 1,\ s = 1,\ p = 2,\ d = 2$
   (c) $k = 3,\ s = 1,\ p = 1,\ d = 1$